Current Status of All Claims in the Application:

1. (Canceled)

2. (Previously Presented) The method of claim 32, wherein the compression of data is performed using a software data compression algorithm.

3. (Original) The method of claim 2, wherein the software data compression algorithm includes one of the following types of algorithms: a zip; a gnuzip; a bzip; a b2zip; a Lempil Ziv; and a LZS (Lempil Ziv Stac).

4. (Previously Presented) The method of claim 32, further comprising successively repeating the receiving and storing of data during the backup window periods and retrieving, compressing and storing compressed data on the backup storage device during successive duty cycles respectively.

5. (Previously Presented) The method of claim 32, wherein the backup storage device is an emulated tape drive containing an array of hard drives.

6. (Previously Presented) The method of claim 32, wherein the data is downloaded over a network from a primary storage location.

7. (Original) The method of claim 6, wherein the data is downloaded over a fiber-channel connection between the primary storage location and the backup storage device.

8. (Original) The method of claim 6, wherein the data is downloaded over an ethernet connection between the primary storage location and the backup storage device.

9. (Original) The method of claim 6, wherein the primary storage location and the backup storage device are part of a storage array network.

10. (Original) The method of claim 6, wherein the primary storage location and the backup storage device are part of a network attached storage configuration.

11. (Currently Amended) The method of claim 32, wherein the backup storage device is directly electrically connected to a server.

12. (Canceled)

13. (Previously Presented) The storage system of claim 31, wherein the controller is further configured to execute a software algorithm to compress the retrieved data.

14. (Previously Presented) The storage system of claim 13, wherein the software algorithm includes one of the following types of algorithms a zip; a gnuzip; a bzip; a b2zip; a Lempil Ziv; and a LZS (Lempil Ziv Stac).

15. (Previously Presented) The storage system of claim 13, wherein the software algorithm is stored in a memory associated with the controller.

16. (Previously Presented) The storage system of claim 31, further comprising a fiber channel controller coupled between the controller and the input/output port which comprises an optical transceiver.

17. (Previously Presented) The storage system of claim 31, further comprising an ethernet controller coupled between the controller and the input/output port which comprises an ethernet transceiver.

18. (Previously Presented) The storage system of claim 31, further comprising a network hub and bridge circuit coupled between the backup storage device and the controller.

19. (Previously Presented) The storage system of claim 31, further comprising:

a primary storage location that allows transmission of uncompressed data from the primary storage location to the backup storage device.

20. (Previously Presented) The storage system of claim 19, wherein the network connection is one of the following types of network connections: fiber channel or ethernet.

21. (Previously Presented) The storage system of claim 19, wherein the primary storage location and the backup storage device are arranged in one of the following: a storage attached network or network attached storage configuration.

22. (Previously Presented) The storage system of claim 19, further comprising a plurality of clients and servers coupled to the primary storage location through a client network.

23-27. (Canceled)

28. (Previously Presented) A storage system comprising:

a primary storage location including an input/output port;

a backup storage device; and

a controller that transmits data between the primary storage location and the backup storage device according to a duty cycle having a predetermined backup window period when uncompressed data from the primary storage location is copied to the backup storage device, and an idle period when uncompressed data from the primary storage location is not being copied in uncompressed form to the backup storage device;

wherein during the idle period the controller retrieves the uncompressed data stored on the backup storage device, compresses the retrieved data, and then re-stores the compressed data on the backup storage device.

29. (Previously Presented) The storage system of claim 28, wherein the compression of data is performed using a software data compression algorithm.

30. (Previously Presented) The storage system of claim 28, wherein the backup storage device is an emulated tape drive containing an array of hard drives.

31. (Previously Presented) A storage system comprising:

a primary storage location including an input/output port;

a backup storage device; and

a controller that copies uncompressed data from the primary storage location to the backup storage device during a predetermined backup period, and

retrieves the uncompressed data from the backup storage device, compresses the retrieved data, and then re-stores the compressed data on the backup storage device during an idle period that begins following a predetermined time period of inactivity through the input/output port.

32. (Previously Presented) A computer-implemented method for storing data from a primary storage location having an input/output port onto a backup storage device, the method comprising the steps of:

copying uncompressed data during a predetermined backup window period from the primary storage location to the backup storage device;

compressing the data with a controller during an idle period defined by when uncompressed data is not being copied from the primary storage location to the backup storage device; and

re-storing the compressed data onto the backup storage device during the idle period.

33. (Previously Presented) The method of claim 32 wherein the step of compressing the data includes beginning the idle period following a predetermined time period of inactivity through the input/output port.

34. (Previously Presented) The method of claim 32 further comprising the step of interrupting the step of compressing the data when activity is detected through the input/output port.

35. (Previously Presented) The method of claim 32 further comprising the step of interrupting the step of re-storing the compressed data when activity is detected through the input/output port.

36. (Previously Presented) A computer-implemented method for storing data from a primary storage location having an input/output port onto a backup storage

device, the method comprising the steps of:

copying uncompressed data from the primary storage location through the input/output port to the backup storage device;

compressing the data copied to the backup storage device with a controller during an idle period that begins following a predetermined time period of inactivity through the input/output port; and

re-storing the compressed data onto the backup storage device with the controller during the idle period.

37. (Previously Presented) The method of claim 36, wherein the compression of data is performed using a software data compression algorithm.

38. (Previously Presented) The method of claim 36, further comprising successively repeating the receiving and storing of data during the backup window periods and retrieving, compressing and storing compressed data on the backup storage device during successive duty cycles respectively.

39. (Previously Presented) The method of claim 36, wherein the backup storage device is an emulated tape drive containing an array of hard drives.

40. (Previously Presented) The method of claim 36, wherein the data is downloaded over a network from a primary storage location.

41. (Previously Presented) The method of claim 36 further comprising the step of interrupting the step of compressing the data when activity is detected through the input/output port.

42. (Previously Presented) The method of claim 36 further comprising the step of interrupting the step of re-storing the compressed data when activity is detected

through the input/output port.